Algoritmica Avanzata – CdL Magistrale in Ingegneria Informatica
Compito, 19/6/2012 (Durata: 2h)

Nome, Cognome, Matricola: _____

Risoluzione di problemi

Si forniscano soluzioni esaurienti e rigorose ai tre problemi seguenti. Gli algoritmi vanno codificati utilizzando lo **pseudocodice** usato in classe. **Attenzione:** Risposte immotivate e prive di prova di correttezza non saranno considerate.

Esercizio 1 [12 punti] Dato un grafo **orientato senza self-loops** $G = (V, E)$, con $V \subseteq \boldsymbol{N}$ e $E \subseteq V \times V - \{(v, v) : v \in V\}$, si vuole determinare un sottoinsieme $E^\star \subseteq E$ di massima cardinalità per cui il sottografo indotto $G' = (V, E^\star)$ risulti essere aciclico. Si fornisca un algoritmo di 2-approssimazione per il problema. (*Suggerimento:* Si osservi che i vertici sono numeri. Si partizionino gli archi di E in due insiemi in modo tale che un ciclo non possa avere tutti gli archi appartenenti allo stesso insieme ...).

Answer: Let $E_\ell, E_g \subseteq E$ with $E_\ell = \{e = (u, v) \in E : u < v\}$ and $E_g = E - E_\ell$. Then, it is easy to show that $G_\ell = (V, E_\ell)$ and $G_g = (V, E_g)$ are both acyclic. For the sake of contradiction, suppose that G_ℓ were not acyclic. Then, there exists a cycle $\langle u_1, u_2, \ldots u_{k-1}, u_k = u_1\rangle$, with $k > 2$ and $(u_i, u_{i+1}) \in E_\ell$, for $1 \leq i < k$. Then, by the definition of E_ℓ we must have:

$$u_1 < u_2 < \ldots < u_{k-1} < u_k = u_1,$$

a contradiction. A similar argument applies to E_g. The algorithm is then the following.

```
APPROX_ACYCLIC(G = (V, E))
E_ℓ ← E_g ← ∅
for each e = (u, v) ∈ E do
    if (u < v) then  E_ℓ ← E_ℓ ∪ {e}
               else  E_g ← E_g ∪ {e}
E' ← argmax{|E_ℓ|, |E_g|}
return E'
```

By virtue of the previous argument, the algorithm returns a subset of edges inducing an acyclic subgraph. Moreover, since $|E_\ell| + |E_g| = |E|$, we have that

$|E'| = \max\{|E_\ell|, |E_g|\} \geq |E|/2 \geq |E^\star|/2$, whence

$$\rho = |E^\star|/|E'| \leq 2$$

. □

Esercizio 2 [10 punti] Si determini l'inverso moltiplicativo di 17 in $\boldsymbol{Z}_{43}^\star$ attraverso l'applicazione dell'algoritmo EXTENDED_EUCLID.

Answer: The chain of recursive calls generated by EXTENDED_EUCLID$(43, 17)$ involves the following instances (in top-down order):

$$(43, 17), (17, 9), (9, 8), (8, 1), (1, 0)$$

which, in turn, return the following Bezout coefficients (in bottom-up order):

$$(1, 0), (0, 1), (1, -1), (-1, 2), (2, -5).$$

We have that $1 = 43 \cdot 2 + 17 \cdot (-5)$, hence $17^{-1} = (-5) \bmod 43 = 38$ in $\boldsymbol{Z}_{43}^\star$. □

Esercizio 3 [11 punti] Data una costante $c > 1$, si determini un limite superiore $m_c(n)$ al numero di estrazioni con reimbussolamento da effettuare da un'urna contenente i primi n numeri naturali che garantisca che il numero medio di numeri distinti estratti sia almeno n/c. Per ogni valore di c fissato, il limite superiore deve essere $m_c(n) = O(n)$.

Answer: Assume that we draw $m_c(n)$ numbers randomly with replacement from the urn. For $1 \leq i \leq n$, let $Y_i^{m^c(n)} = 1$ if number i has not been extracted after the $m^c(n)$ trials, and 0 otherwise. Clearly,

$$\Pr(Y_i^{m^c(n)} = 1) = (1 - 1/n)^{m^c(n)} < e^{-m^c(n)/n}.$$

Let now X be the random variable denoting the number of distinct numbers observed after the $m^c(n)$ extractions. Clearly $X = n - \sum_{1 \leq i \leq n} Y_i^{m^c(n)}$ and $E[X] = n - \sum_{1 \leq i \leq n} E[Y_i^{m^c(n)}] > n(1 - e^{-m^c(n)/n})$. Observe then that it suffices to make sure that

$$\begin{aligned} & 1 - e^{-m^c(n)/n} \geq 1/c \\ \Leftrightarrow \quad & (1 - 1/c) \geq e^{-m^c(n)/n} \\ \Leftrightarrow \quad & m^c(n) \geq n \ln(c/(c-1)). \end{aligned}$$

We then choose $m^c(n) = \lceil n \ln(c/(c-1)) \rceil$ and note that $m^c(n) = O(n)$ for any fixed constant $c > 1$.

□

Algoritmica Avanzata – CdL Magistrale in Ingegneria Informatica

Compito, 24/6/2014 (Durata: 2h)

Nome, Cognome, Matricola: _____

Si forniscano soluzioni motivate e rigorose ai tre problemi seguenti.

Esercizio 1 [11 punti] Dato un grafo non orientato $G = (V, E)$, con $E \neq \emptyset$, un *dominating set* di G è un sottoinsieme $D \subseteq V$ tale che:

$$\forall v \in V : (v \in D) \vee (\exists u \in D : \{u, v\} \in E).$$

In altre parole, un generico nodo o è in D oppure è adiacente a un nodo di D. Si vuole determinare il dominating set $D^\star$ di cardinalità minima.

Sia Δ il grado massimo di un nodo in G. Si fornisca un algoritmo APX_DS di approssimazione per il problema con $\rho_{\text{APX\_DS}} = O(\ln(\Delta + 1))$.

(*Suggerimento:* si cerchi di applicare l'algoritmo di approssimazione visto in classe per SET-COVER.)

Answer:

Given $G = (V, E)$, for each $v \in V$ we define its *extended neighborhood* $N'_v = \{u \in V : (u, v) \in E\} \cup \{v\}$, and observe that $|N'_v| \leq \Delta + 1$. Let now $\mathcal{F} = \{N'_v : v \in V\}$ be the family of extended neighborhoods. Then, $(V, \mathcal{F})$ is an instance of the SET-COVER problem. Observe that any feasible solution of $(V, \mathcal{F})$, say $\mathcal{S} = \{N_{v_1}, \ldots, N_{v_k}\}$ corresponds to the dominating set $D_{\mathcal{S}} = \{v_1, \ldots, v_k\}$ and vice versa, since the extended neighborhoods of the nodes in $D_{\mathcal{S}}$ cover the whole vertex set. Moreover, the cost of $\mathcal{S}$ and $D_{\mathcal{S}}$ is clearly the same.

In the light of the above observations, algorithm APX_DS($G = (V, E)$) simply runs the approximation algorithm APPROX-SET-COVER($V, \mathcal{F}$) seen in class and returns the set $D_{\mathcal{S}}$ associated to the approximate solution $\mathcal{S}$ returned by the latter algorithm. (The pseudocode is a trivial variation of APPROX-SET-COVER and is omitted for brevity.) We obtain that

$$\begin{aligned}
\rho_{\text{APX_DS}} &= \frac{|D_{\mathcal{S}}|}{|D^\star|} \\
&= \frac{|\mathcal{S}|}{|\mathcal{S}^\star|} \\
&\leq H(\max\{|S| : S \in \mathcal{F}\} \\
&= H(\max\{|N'_v| : v \in V\}
\end{aligned}$$

$$\begin{aligned} &\leq H(\Delta+1) \\ &= O(\ln(\Delta+1)). \end{aligned}$$

□

Esercizio 2 [10 punti] Si consideri un criptosistema a chiave pubblica di tipo RSA in cui il dominio dei messaggi è $\boldsymbol{Z}_n$, con $n = 11 \times 23 = 253$. Data la chiave pubblica $P = (7, 253)$ determinare la corrispondente chiave segreta S e la relativa funzione di decodifica $S(M)$ ad essa associata.

Answer: We have that $\phi(n) = (11-1)\cdot(23-1) = 220$. Given the public key $P = (7, 253)$, in order to obtain the corresponding secret key, we must determine the multiplicative inverse of 7 in $\boldsymbol{Z}^{\star}_{220}$. In turn, such an inverse can be obtained by applying Algorithm EXTENDED EUCLID (EE) to determine the Bezout equality between 1=gcd(220, 7), 220 and 7. On $(220, 7)$, EE performs calls on $(7, 3)$, $(3, 1)$, and finally $(1, 0)$. From bottom up, the values returned by these calls are $(1, \{1, 0\})$, $(1, \{0, 1\})$, $(1, \{1, -2\})$ and finally $(1, \{-2, 63\})$. It follows that $S = (63, 253)$, hence, for any message $M \in \boldsymbol{Z}_{253}$, $S(M) = M^{63} \bmod 253$. □

Esercizio 3 [11 punti] Il problema MAX-3-CNF-OVERSAT richiede di determinare, data una formula $\langle \Phi(x_1, \ldots, x_n) = C_1 \wedge \ldots \wedge C_m \rangle$ in formato 3-CNF, il massimo numero di clausole che contengano almeno due letterali veri sotto un fissato assegnamento di verità. Si fornisca un semplice algoritmo di approssimazione randomizzato $A(\langle \Phi \rangle)$ per MAX-3-CNF-OVERSAT e se ne studi il fattore di approssimazione $OPT/E[A(\langle \Phi \rangle)]$.

Answer: We propose the following simple randomized algorithm:

```
A(⟨Φ(x)⟩)
n ← x.len
⋆ Let Φ(x) = ⋀_{i=1}^{m} C_i ⋆
for j ← 1 to n do b_j ← RANDOM({0,1})
count ← 0
for i ← 1 to m do
    if (⋆C_i(b) features two true literals ⋆)
        then count ← count + 1
return count
```

Let $X_i = 1$ if the i-th clause contains two true literals under the random assignment $\boldsymbol{b}$. Then

$$E[A(\langle \Phi \rangle)] = E[count] = E[\sum_{i=1}^{m} X_i] = \sum_{i=1}^{m} E[X_i] = \sum_{i=1}^{m} \Pr(X_i = 1).$$

Since there are 4 over 8 configurations of three variables that feature at least two true literals in a clause, we have $\Pr(X_i = 1) = 1/2$ for $1 \leq i \leq m$, whence $E[A(\langle\Phi\rangle)] = m/2$. It follows that

$$\rho_A = OPT/E[A(\langle\Phi\rangle)] \leq m/(m/2) = 2.$$

□

Algoritmica Avanzata – CdL Magistrale in Ingegneria Informatica

Compito, 15/9/2014 (Durata: 2h)

Nome, Cognome, Matricola: _____

Si forniscano soluzioni motivate e rigorose ai tre problemi seguenti.

Esercizio 1 [11 punti] Si considerino istanze non vuote $\langle S, t\rangle$ del problema di ottimo relativo a SUBSET SUM tali che $\forall s \in S : s \leq t$. Si consideri il seguente algoritmo che prende in ingresso una di tali istanze:

```
APPROX_OPT_SS(⟨S, t⟩)
{s_1, s_2, ..., s_n} ← SORT-DECREASING(S)
sum ← s_1
i ← 2
while ((i ≤ n) and (sum + s_i ≤ t)) do
    sum ← sum + s_i
    i ← i + 1
return sum
```

Nel codice, SORT-DECREASING(S) rappresenta una chiamata a una routine di ordinamento che ritorna gli elementi dell'insieme S in ordine **decrescente**. Si dimostri che APPROX_OPT_SS è un algoritmo di ρ-approssimazione per le istanze di SS sopradefinite, con $\rho < 2$.

Answer: Since sum is initialized to $s_1 \leq t$ and an element $s_i \in S$ is added to sum only if $sum + s_i \leq t$, the value returned is indeed the cost of a feasible solution. Let $s^\star$ be the optimal cost for the instance. It suffices to show that the value sum returned is such that $s^\star/sum \leq 2$, which immediately implies that APPROX_OPT_SS is a 2-approximation algorithm.

Consider first the case when, on exit from the while loop, it is $i = n + 1$. Then the algorithm returns $sum = \sum_{s\in S} s$, which clearly implies that $sum = s^\star$, whence $\rho = s^\star/sum = 1 < 2$. Otherwise, on exit from the while loop, it is $i \leq n$, and $sum + s_i > t$. Observe that , $s_i < s_1$ by the sorting, and $s_1 \leq sum$ since s_1 is always added to sum. Therefore we obtain $t < sum + s_i < sum + s_1 \leq 2 \cdot sum$, which implies $sum > t/2$. Hence $\rho = s^\star/sum < t/(t/2) = 2$. □

Esercizio 2 [10 punti] Si consideri un criptosistema a chiave pubblica di tipo RSA in cui il dominio dei messaggi è $\boldsymbol{Z}_n$, con $n = 7 \times 19 = 133$. Data la chiave pubblica $P = (7, 133)$ determinare la corrispondente chiave segreta S e la relativa funzione di codifica $S(M)$ ad essa associata.

Answer: If $n = 7 \times 19 = 133$, we have that $\phi(n) = 6 \times 18 = 108$. The given public key is $P = (7, 133)$, whose associated encryption function is $P(M) = M^7 \bmod 133$, for each $M \in \boldsymbol{Z}_{133}$. Under RSA, the corresponding secret key will be (d, n), where d is the multiplicative inverse of 7 in $\boldsymbol{Z}^{\star}_{108}$. The value of d can be inferred by writing 1=gcd(108,7) as an integer linear combination of 108 and 7 (Bezout's equality). The coefficients of such combination are obtained by applying EXTENDED_EUCLID (EE) to 108 and 7. We obtain the following recursive calls:

EE(108,7) $\rightarrow$ EE(7,3) $\rightarrow$ EE(3,1) $\rightarrow$ EE(1,0)

whose return values (in inverse order) are:

$\{1, (1, 0)\} \rightarrow \{1, (0, 1)\} \rightarrow \{1, (1, 0 - 2 \cdot 1)\} \rightarrow \{1, (-2, 1 - (15) \cdot (-2))\}$ $= \{1, (-2, 31)\}$

Since $31 \cdot 7 = 1 + 2 \cdot 108$, we have that in $\boldsymbol{Z}^{\star}_{108}$, $d = e^{-1} = 31$, hence $S = (31, 133)$ and $S(M) = (M^{31} \bmod 133)$ for each $M \in \boldsymbol{Z}_{133}$. □

Esercizio 3 [11 punti] Il problema MAX-3-CNF-FULLSAT richiede di determinare, data una formula $\langle \Phi(x_1, \ldots, x_n) = C_1 \wedge \ldots \wedge C_m \rangle$ in formato 3-CNF, il massimo numero di clausole che abbiano tutti i letterali veri sotto un fissato assegnamento di verità. Si fornisca un semplice algoritmo di approssimazione randomizzato $A(\langle \Phi \rangle)$ per MAX-3-CNF-FULLSAT e se ne studi il fattore di approssimazione $OPT/E[A(\langle \Phi \rangle)]$.

Answer: We propose the following simple randomized algorithm:

```
A(⟨Φ(x)⟩)
n ← x.len
⋆ Let Φ(x) = ⋀_{i=1}^{m} C_i ⋆
for j ← 1 to n do b_j ← RANDOM({0,1})
count ← 0
for i ← 1 to m do
    if (⋆C_i(b) features three true literals ⋆)
        then count ← count + 1
return count
```

Let $X_i = 1$ if the i-th clause contains two true literals under the random assignment $\boldsymbol{b}$. Then

$$E[A(\langle \Phi \rangle)] = E[count] = E[\sum_{i=1}^{m} X_i] = \sum_{i=1}^{m} E[X_i] = \sum_{i=1}^{m} \Pr(X_i = 1).$$

Since there is 1 over 8 configurations of three variables that feature at least two true literals in a clause, we have $\Pr(X_i = 1) = 1/8$ for $1 \leq i \leq m$, whence

$E[A(\langle\Phi\rangle)] = m/8$. It follows that

$$\rho_A = OPT/E[A(\langle\Phi\rangle)] \leq m/(m/8) = 8.$$

□